# The Evaluation of GPU-Based Programming Environments for Knowledge Discovery

***John Johnson, Randall Frank,*** and ***Sheila Vaidya***
Lawrence Livermore National Labs
Phone: 925-424-4092
Email Addresses: {jjohnson, fjfrank, vaidya1}@llnl.gov

## Introduction

Revolutionary advances in computer graphics technologies, driven by the needs of 3D gaming, have resulted in specialized SIMD floating point rendering engines known as GPUs. These GPUs are programmed via graphics libraries such as OpenGL, but have very general programming architectures. These cards are handily exceeding Moore's law performance predictions and are expected to continue to do so for some time. The size and cost competitive nature of the gaming industry combine to make these systems extremely affordable. Today, GPUs with over 40GF can be bought for around $300 and they are expected to increase to around 1000GF for about that same cost by the 2005 timeframe. These systems form the core of distributed interactive systems but can also be applied to many processes other than rendering and visualization. At present, the non-visualization uses of these systems have been limited to classically streaming or vector floating point bound processes.

We will present early results in the use of these GPU systems to perform computations on alternative types of algorithms that are not traditionally FLOP bound, such as those utilized in video image processing, text processing and semantic graph traversal and analysis. Knowledge discovery based application areas should, minimally, benefit from the extreme memory bandwidths present on GPU systems (over 23GB/sec in current systems), and are in a position to exploit the FLOP rich GPU environment to enhance the fidelity and complexity of their computation. Some of our early studies have already shown orders of magnitude performance speedup for specific applications.

## The GPU Based Compute Platform

Two advantages of GPUs are their extremely high memory bandwidth and their unique gather capabilities. We are investigating applications that exploit both of these features. As a key first step we are investigating the mapping of data onto the current GPU architectures via pointer-less indirection techniques and implicit parallel storage techniques. The design is expected to draw from recent work on tiled, paged boundary conditions on GPU systems. The initial targets are temporal image processing algorithms commonly used in the processing of data like surveillance video and facial biometrics. Basic algorithms for filtering and feature detection and tracking are being implemented and demonstrated to apply to large, parallel data streams.

One of the difficulties in the scaling and parallelization of algorithms on GPU systems stems from the very nature of the data structures. Following the image processing work, will focus on non-traditional HPC data structures. In the next several months we expect to investigate the

# Report Documentation Page

*Form Approved*
*OMB No. 0704-0188*



| 1. REPORT DATE | 2. REPORT TYPE | 3. DATES COVERED |
|---|---|---|
| **01 FEB 2005** | **N/A** | **-** |

| 4. TITLE AND SUBTITLE | 5a. CONTRACT NUMBER |
|---|---|
| **The Evaluation of GPU-Based Programming Environments for Knowledge Discovery** | 5b. GRANT NUMBER |
| | 5c. PROGRAM ELEMENT NUMBER |
| 6. AUTHOR(S) | 5d. PROJECT NUMBER |
| | 5e. TASK NUMBER |
| | 5f. WORK UNIT NUMBER |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Lawrence Livermore National Labs** | 8. PERFORMING ORGANIZATION REPORT NUMBER |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) | 10. SPONSOR/MONITOR'S ACRONYM(S) |
| | 11. SPONSOR/MONITOR'S REPORT NUMBER(S) |

12. DISTRIBUTION/AVAILABILITY STATEMENT
**Approved for public release, distribution unlimited**

13. SUPPLEMENTARY NOTES
**See also ADM00001742, HPEC-7 Volume 1, Proceedings of the Eighth Annual High Performance Embedded Computing (HPEC) Workshops, 28-30 September 2004 Volume 1., The original document contains color images.**

14. ABSTRACT

15. SUBJECT TERMS

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | |
| **unclassified** | **unclassified** | **unclassified** | **UU** | **19** | |

applicability of GPU system to string and list processing functions. These have been difficult to map onto streaming processing systems, but recent advances in pixel shader technology would suggest that it may be possible to perform hundreds of parallel text searches in parallel in a streaming, multi-pass GPU architecture. We intend to exploit similar advances in texture fetching operations to investigate the use of GPUs in pointer-less list searching and comparison problems. These research advances hold the potential of allowing these systems to be applied to other data mining problems and the processing of transaction orientated data, such as the analysis of web traffic or semantic graphs.

The GPU enhanced system is not a static target and tremendous advances are announced on nearly a bi-annual basis by vendors. Additionally, it will be useful to compare results from these GPU based systems with the results from other architectures that are being developed in parallel (e.g. BlueGene/L and Merrimac). In parallel with the basic algorithmic efforts, we are performing research into the integration of this work with higher level semantic languages with multiple system targets. The integration this work, in particular the non-traditional data structures for strings and lists into streaming languages such as Brook will allow the work to target a number of other real or simulated architectures. As a result, virtual performance comparisons can be made with these architectures. As efforts in this space progress, the model will be adapted to next generation graphics architectures such as upcoming future architectures such as the proposed “Cell” based systems.

Abstract

Agenda

# The GAIA Project:
## Evaluation of GPU-Based Programming Environments for Knowledge Discovery

*Jeremy Meredith*

David Bremer, Lawrence Flath, John Johnson, Holger Jones, Sheila Vaidya, Randall Frank*

Lawrence Livermore National Laboratory

UCRL-PRES-206819

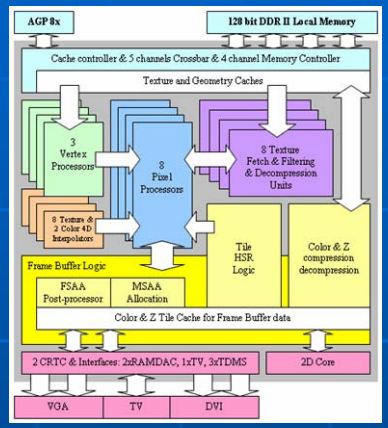
This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

# Motivation

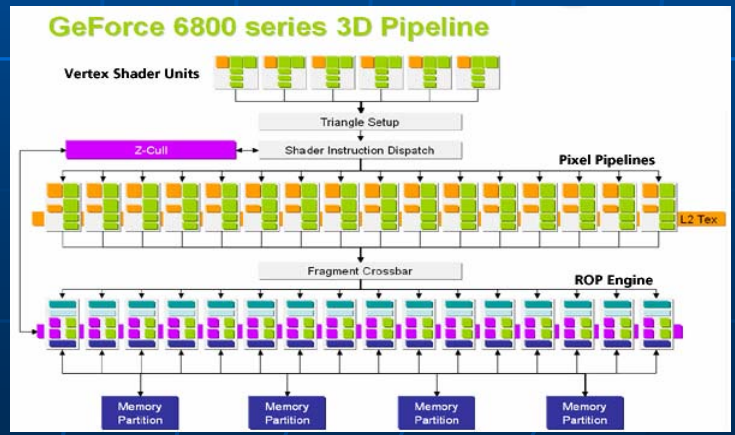
- Trends in the graphics marketplace
  - Inherent parallelism of graphics tasks
  - Performance increasing faster than for CPUs
  - Move to programmable hardware
  - Effects of mass markets
- Not expected to end anytime soon…
  - Today: 40GF, 2GB/s I/O, 30GB/s memory
  - 2006: 100GF, 8GB/s I/O, 60GB/s memory
  - 2007: 1TF…

# The NV40 and the Sony Playstation 3

- Are graphics trends a glimpse of the future?
- The nVidia NV40 Architecture
  - 256MB+ RAM
  - 128 32bit IEEE FP units @ 400Mhz
  - 220M transistors, 110W of power
- The PlayStation3 (patent application)
  - Core component is a cell
    - 1 “PowerPC” CPU + 8 APUs (“vectorial” processors)
    - 4GHz, 128K RAM, 256GFLOP/cell
  - Multiple cells (Phone, PDA, PS3, ...)
    - Four cell architecture (1TFLOP)
    - Central 64MB memory
- Keys
  - Streaming data models
  - Cache-driven/cache-oblivious computing

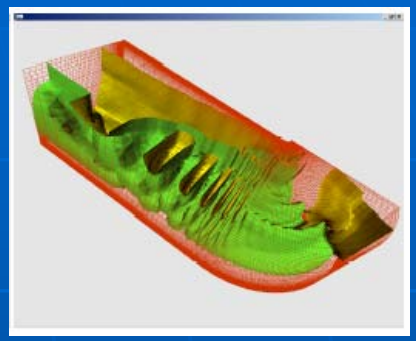



nVidia NV30

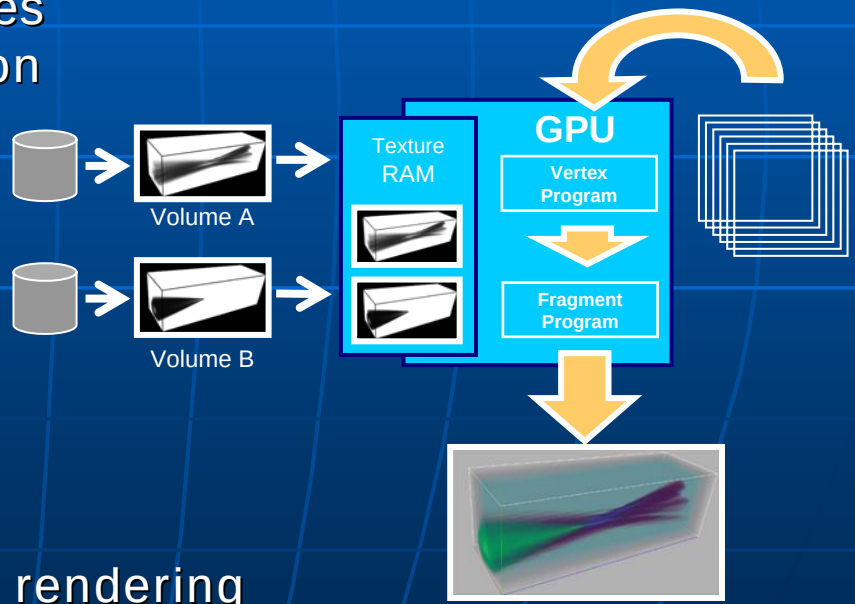



nVidia NV40

# Data representations for GPUs



- Programmable FP SIMD engines, 40-100GF today, 1TF by ’06
- Where can they be exploited?
  - Many advantages for the data pipeline
  - Data/algorithmic design challenges
  - Possible applicability for simulation
  - Many current research projects on scientific computing, databases, audio processing
- Current projects
  - Programmable rendering pipeline
    - Multi-variate, interactive
    - Increased graphics precision
  - Image composition pipeline
  - Implementation of physics based rendering
    - Simulated radiography, diffraction computation
  - Large image geo-registration
    - 100x performance improvement over CPU



# Specific Project Goals

- Investigate use of COTS technologies for computation
  - "Non-traditional" applications
    - Image and speech
    - String, statistical, graph…
  - Mechanisms necessary for exploitation
    - Data infrastructure (e.g. cache coherent streaming…)
    - Software abstractions
  - Delineate some boundary conditions on their use
    - Evaluation vs CPU based solutions
    - Parameter-space investigation

# Data Infrastructure

- Forms the basis of a comparative framework
  - Support both GPU and CPU algorithmic implementations
  - Targets multiple platforms
  - Provides data abstraction
    - “Tile-based” streaming
    - Cache coherency control
    - CPU to GPU to CPU glue layer
  - Utilizes higher-level languages for algorithms
    - Cg, Brook, GLSL, etc

# Image Processing Applications

- Common attributes
  - Large, streaming imagery on a single gfx card
  - Parallel 1D and 2D applications
  - Multi-spectral (four, possibly temporal channels)
- Discrete convolution
  - Arbitrary kernels
- Correlation
  - Separate threshold, search, and detection phase included

# String Processing Applications

- Representation and bandwidth characteristics
- String comparison
  - “Bulk” comparison operations individual outputs
- String sorting
  - Based on string comparison
  - Batched sort based on radix algorithms
- String searching
  - “Wildcard” pattern matching
  - Sort-based element search

# Other Application Targets

- Image transforms
  - FFT, Wavelet
  - Many application domains
- Statistical functions on images
  - Moments, regression (general linear model)
  - Hypothesis/model driven image processing, texture characterization, etc
  - Hidden Markov Models
- Graph search
  - Structured (fully connected) or unstructured graphs, detect and return lowest cost path
  - Many application domains

# System Targets

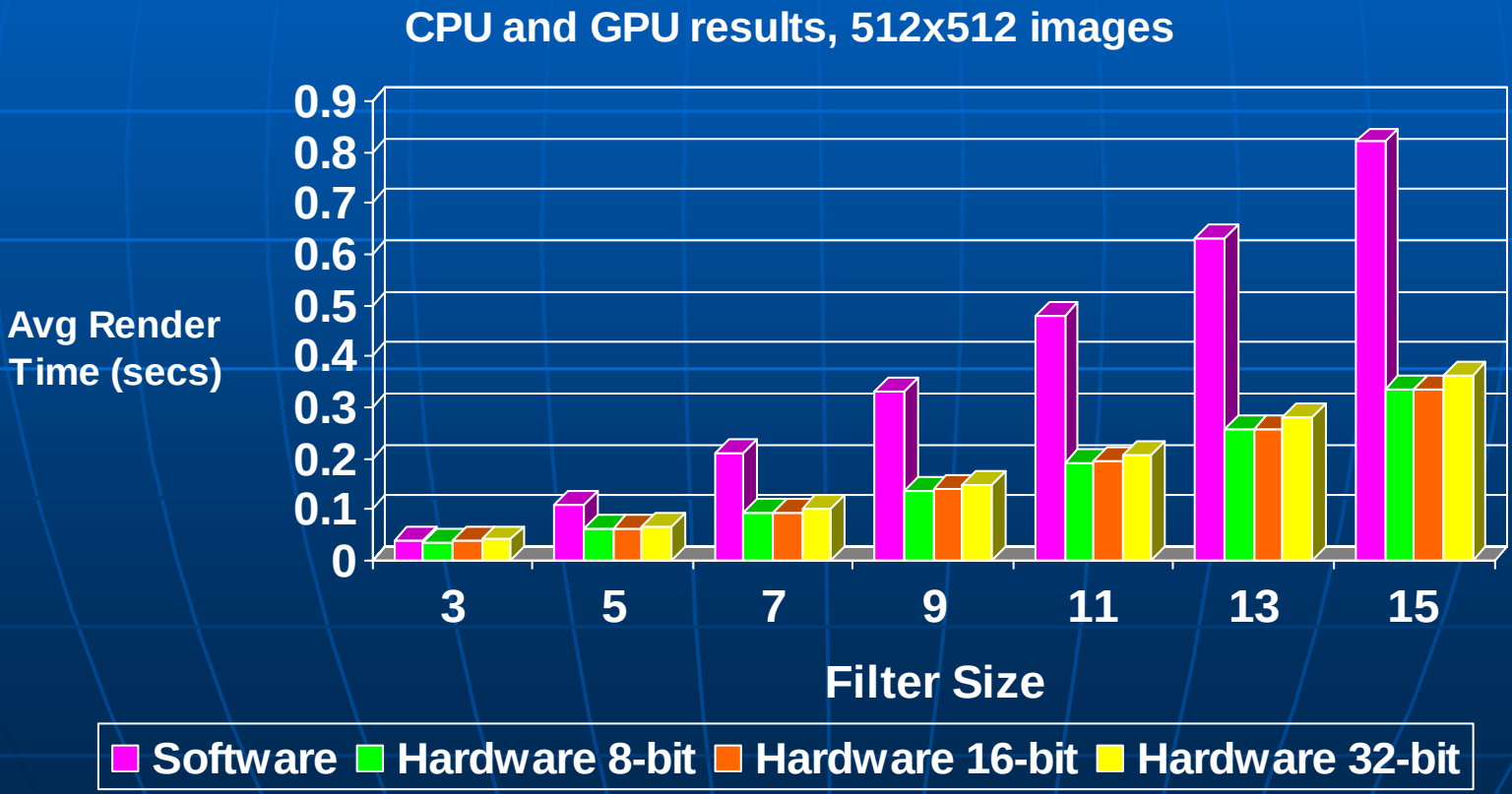
- Constrained system targets based on resource limits
- Hardware targets
  - nVidia: NV3x, NV4x, NV5x
    - Focus on NV4x due to new branching capabilities
    - Dual CPU IA32 platform
    - PCI-Express (PCIe) enhanced readback and async bandwidth
  - BG/L and Merrimac
- OS targets
  - Primarily Linux, some Windows due to driver issues
- Language targets
  - nVidia Cg, Brook

# Convolution Timing Results

- All timings count download, render, and readback
- First render pass is excluded from the count
- Overhead to load shader can be substantial

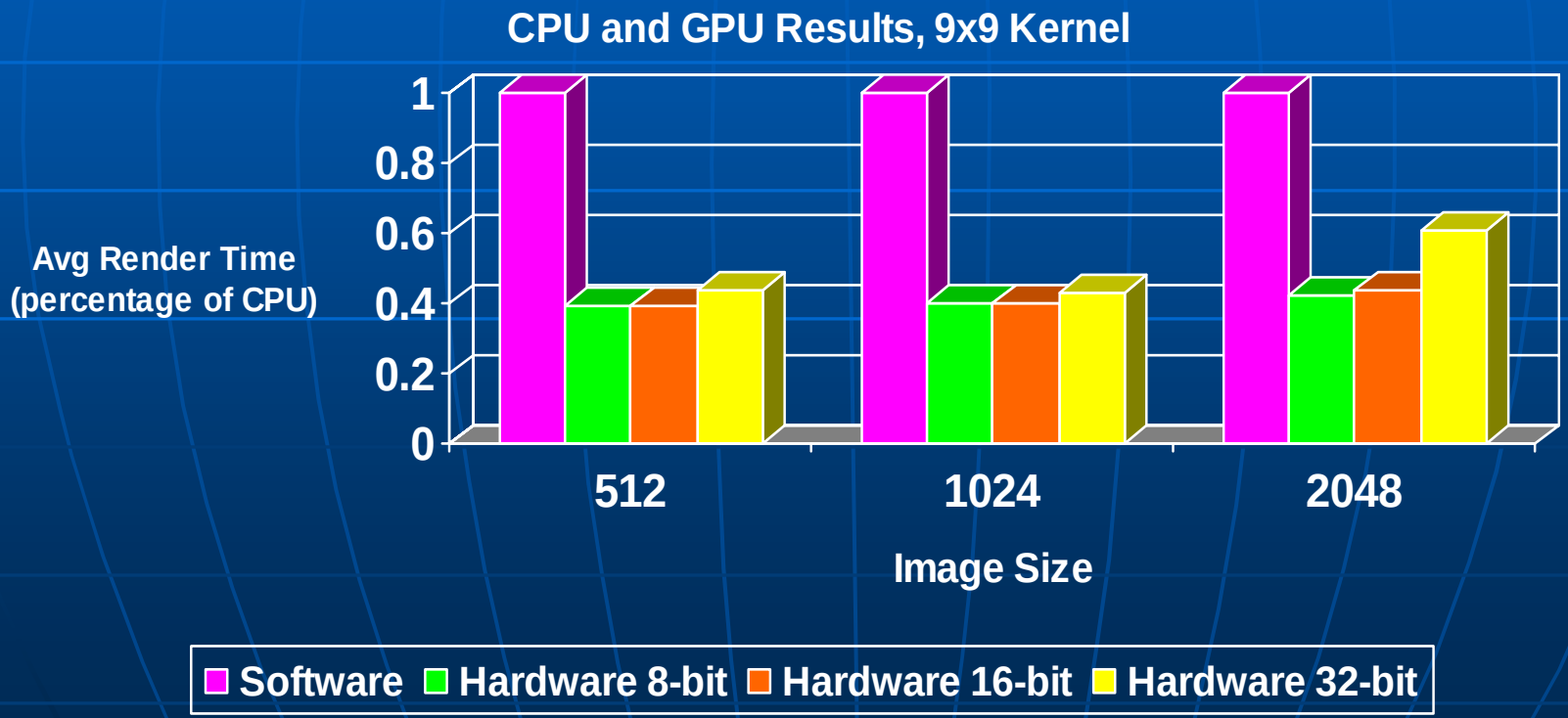
# Convolution Timing Results

- Software vs. two-texture hardware implementation
- At all but the smallest kernel sizes, GPUs are much faster



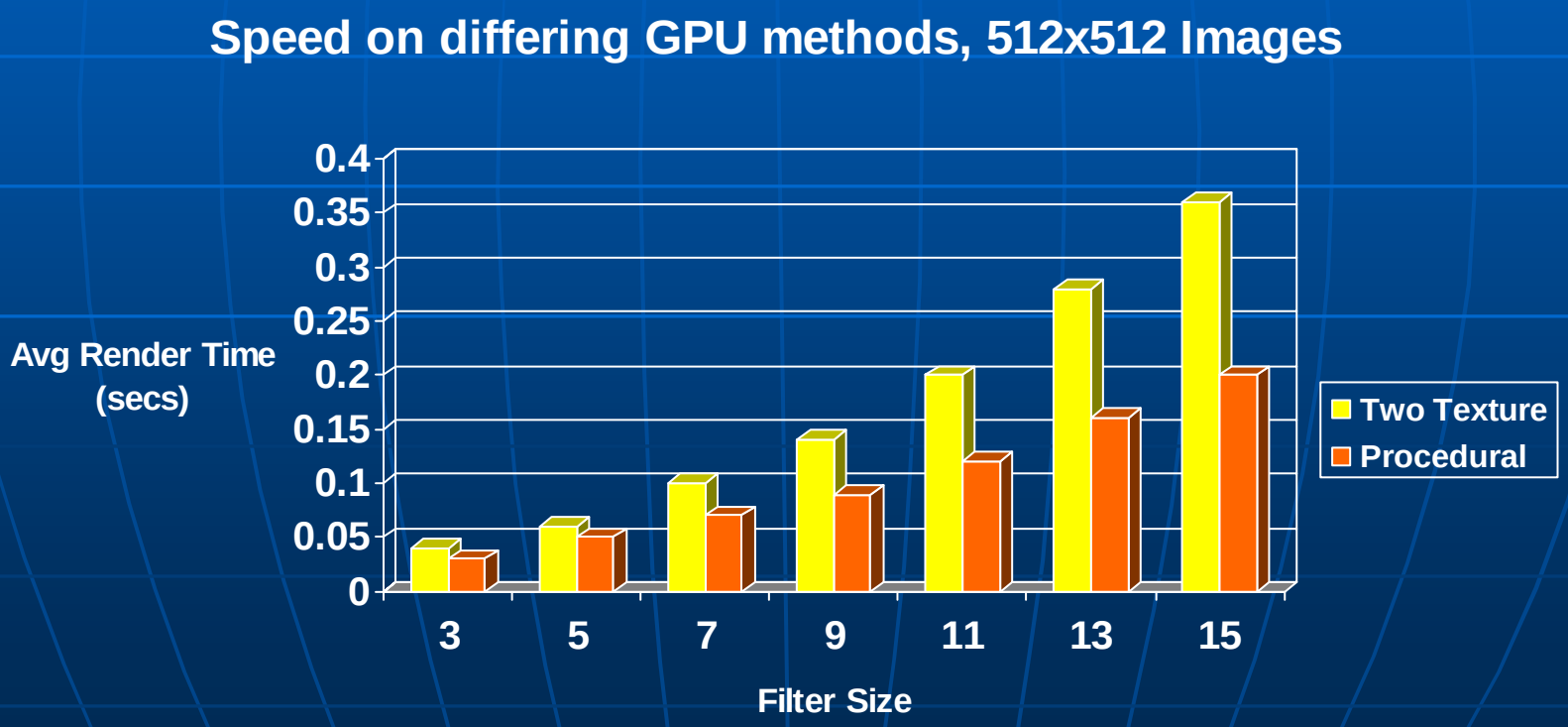
# Convolution Timing Results

- Software vs. two-texture hardware implementation
- 32-bit textures use more memory bandwidth



# Convolution Timing Results

- Two-texture vs. procedural hardware implementations
- Two-texture implementation requires more memory bandwidth



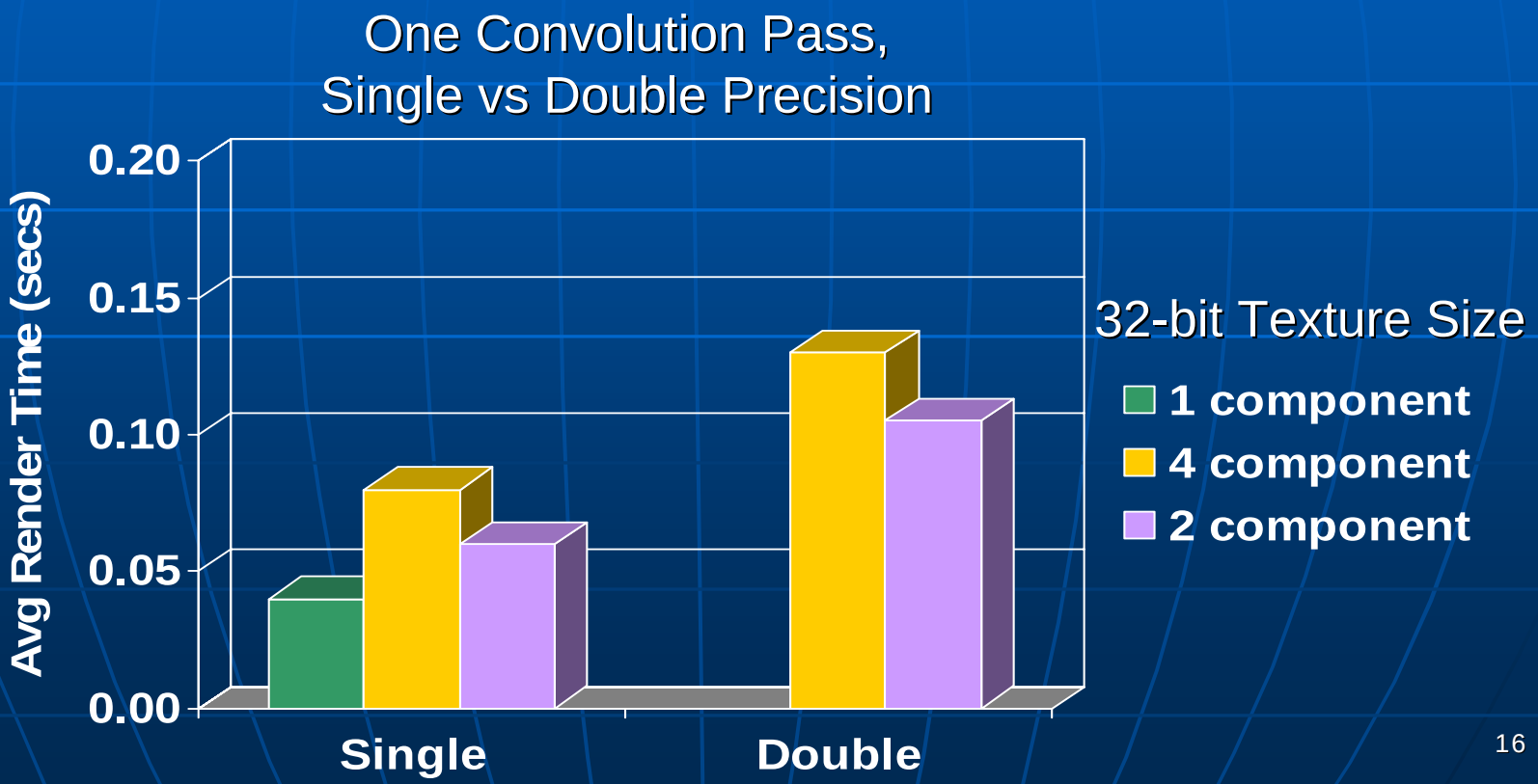
# Double Precision

- Port of David Bailey's *single-double* Fortran library* to NVidia's Cg language
- Can emulate double precision
- Use two single-precision floats
- High order float is estimate to the *double*; Low order float is error of that estimate
- Resulting precision is almost *double*
- The exponent remains at *single* range

available at htpp://crd.lbl.gov/~dhbailey/mpdist

# Double Precision Results

- Convolution with single and emulated-double arithmetic
- Double precision only 1.5x slower than single precision at the same texture depth



# Future Plans

- Obtain results for a variety of algorithms including strings, HMMs, and FFTs
- Include performance and accuracy
- Extend to new architectures as available (e.g. Merrimac)
- Explore other high-level languages (e.g. brook implementations and other streaming languages)
- Launch a benchmarking web site: http://www.llnl.gov/gaia